

EELS: An Embedded E-Learning Solution

Christopher Bailey (c.bailey@bristol.ac.uk)¹, Peter Walker (peter.walker@bristol.ac.uk)¹, Simon Price (simon.price@bristol.ac.uk)¹, Bevis Miller (bev.miller@bristol.ac.uk)²

¹Institute for Learning & Research Technology, University of Bristol

²Department of Clinical Veterinary Science, University of Bristol

1 EXECUTIVE SUMMARY

Within most undergraduate medical disciplines, students spend time attending off-site placements or rotations in a clinical environment. To access this clinical experience, students are required to submit a report that documents the progression of a patient. This rotation is a logistic challenge for tutors attempting to keep track of their students' progress. Our goal was to design a system students could use for submitting their case reports online and that tutors could use to review and provide feedback. We wanted to replace the traditional end-of-rotation, monolithic report with a completely web-based alternative, providing students with a richer, more expressive environment for documenting cases, and giving tutors a centralised location for tracking progress, managing students and marking their case reports.

To this end we have developed EELS, the Embedded E-Learning System; a highly flexible case reporting system. The software is built using open source XForms technology to manage the forms for data entry. XForms provides a number of key benefits which we harness, such as easy form validation and constraint checking, support for repeated sections in forms, and Web 2.0 (AJAX) style interactions such as auto-saving. At the heart of our system is a custom-built dynamic workflow engine which manages the full lifecycle of cases, from their creation, through edit/review cycles, to assessment, archival and publishing. In addition to defining the stages and terminology, we can also fine tune access rights at every stage; with a group-based permissions component specifying who can edit, view, review or search for cases. We have also harnessed the data abstraction features of XForms to develop a WYSIWYG editor allowing non-technical authors to design and generate new case forms.

The EELS system brings a number of key pedagogical improvements to existing practice:

- Providing a library of exemplar cases
- Enhancing discourse between staff and students
- Supporting reflective & life long learning

One of the key benefits of our innovation is that students are encouraged to document cases in much more depth than previously possible. Allowing users to upload multimedia material means that for the first time students can attach photographs, video files, audio samples, consent forms, x-rays images etc., which help to create a much more comprehensive case report. We have also implemented a drawable region which allows students to create their own drawings or annotate existing images such as, for example, marking regions on a dental chart.

The flexibility of our dynamic workflow engine, form designer and use of XForms technology also means that our system has wide applicability, not just within medical domains. It is easily configurable to any discipline where structured data needs to be obtained from students - this could be recording consent forms, checking skills progression, documenting reports, submitting coursework or running course evaluations.

2 PROJECT BACKGROUND

The EELS project was instigated in late 2005 by Dr Bevis Miller, Senior Lecturer in Animal Husbandry at the University of Bristol. It arose from an earlier prototype application that enabled Veterinary

students to record “clerking cases” - animal examinations and diagnoses - via the Web.

The project was proposed as having three distinct technical aims

1. A Web-based mechanism to record student case notes (placement or field experience) and for tutors/supervisors to be able to review and comment. Tutors would be able to augment the details of the best case to serve as examples and/or a source of data for online tutorials.
2. Advanced search mechanisms to retrieve case details - either by the students or staff involved in the case or by others who wanted to learn from the experience of others.
3. Tools to build on the search and allow easy integration of case details within web-based tutorials. It was an early decision to propose tools to integrate with third-party tutorials packages to allow maximum flexibility.

A high-level Steering Group was convened to manage the project. This group included senior academics from all University faculties. Members of the group saw potential value to students and staff in their departments. Funding for two years was secured and this was later extended for a third year. Need for the software was greatest within the medical disciplines so it was decided to pilot the project here.

The project has stuck to the aims outlined above with the first two being prominent to date. Since early 2007 Vet students and tutors have been using the system - termed the “Clinical Experience Recorder” (CER). Dental students and staff began using the system later in 2007. Human Medicine plan to employ EELS later this year and Engineering are considering use.

The Veterinary school has now employed new staff specifically with the role of building online tutorials from the case materials built to date.

The flexibility of the EELS solution has been proved by its rapid adaptation to another important area of e-learning - student “portfolios” of experience and learning. A derivation of the CER was created to allow both Veterinary and Dental students to pilot e-portfolio screens tailored to their specific disciplines and needs.

3 PEDAGOGICAL ASPECTS

The primary pedagogical aim of the project was to enhance the learning students gained during their clinical practice. This consisted of several sub aims. These were:

1. To enhance student reflection upon their own clinical cases by capturing all aspects of the case on the web using multimedia tools, the addition of multimedia greatly enhancing the quality of the case detail. By the use of structured web forms, students record case progression; from presenting case history through to prognosis (Figure 1) at each step on the decision taken.
2. Sharing via the web. Cases can be viewed and followed by other students to maximise the use of this limited clinical resource. The project also aims to explore the pedagogic gain of peer review and feedback by students of one another's cases.
3. Tracking Case Development. Often case reports follow a patient over weeks or months, we envisage tutors being able to highlight interesting cases as they arrive and then allow other students to follow their progress. This allows a student to experience the case *as it develops* which would be very difficult to emulate inside a static report. Learning from mistakes, dead-ends, reformulating diagnosis or gathering new information helps students formulate better models for experiential learning.
4. To enable online communication between students and their tutors. Students can submit their cases for comment and advice by the supervising clinicians as many times as each finds acceptable. The case is then signed off by the students and assessed by the clinician. This enables rapid detailed formative and summative feedback.
5. Completed cases can be archived in a fully searchable database. These then provide a

valuable learning resource in their own right. Both staff and students can review the cases either as part of set work or as background research into a specific clinical condition. The library of clinical cases also will provide an invaluable resource of clinical material which can provide the core information for the development of an on line clinical assessment tool. This will involve a multi-layered tree structure requiring decision making at several levels. Similar to many commercial puzzle-based games - access to higher levels will only be possible once the level the student is on has been successfully completed. Separate rules will apply in formative versus summative settings so that formative learners will be allowed to progress.

6. To record accredited achievement of specific “day one” clinical skills. The accreditation process involving veterinary qualified staff both on and off site (veterinary practice clinical placements) signing off student claims. This aims to increase the number of assessment opportunities for students (e.g. off site placements) and to provide them with rapid feedback on their clinical skills progress.

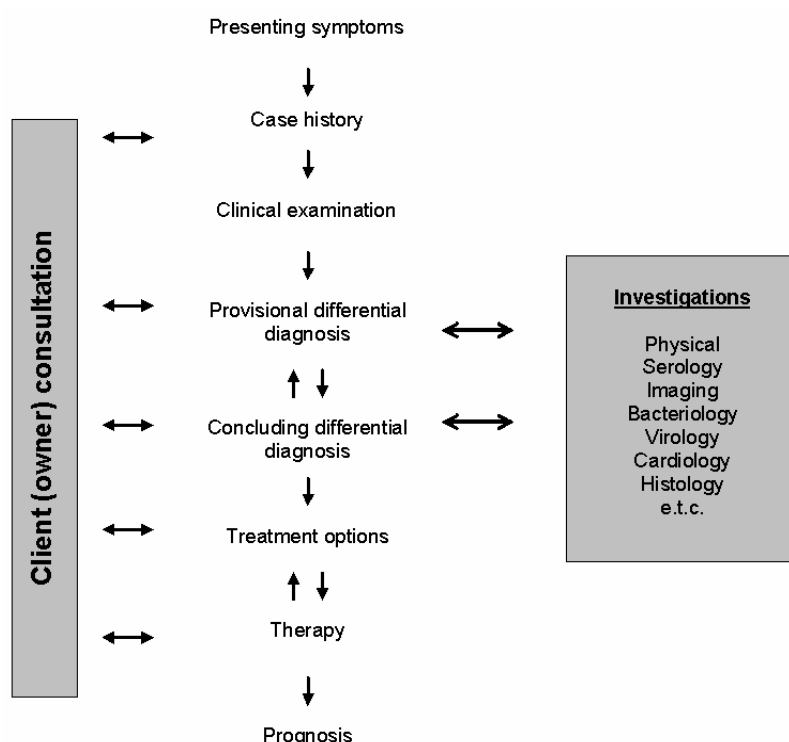


Figure 1 - Stages of a clinical animal case

The archive of quality clinical case will also contribute to a secondary aim of the project, which is to support staff’s own clinical research.

4 TECHNICAL SPECIFICATIONS

In the early stages of system design we chose to adopt a new emerging web standard called XForms, developed by the W3C1. XForms can be viewed as a replacement to the <FORM> element seen in traditional HTML documents. Its utility and power comes from a conscious effort to create a distinction between the data being recorded, the presentation of the form elements on the web page and the constraints and rules that limit the format of the data. This architecture follows the

¹ The Forms working Group, W3C. URL: <http://www.w3.org/MarkUp/Forms/>

MVC or Model View Controller design pattern. The first version of XForms, 1.0, became a W3C recommendation in October 2003 and an updated version, 1.1, is currently a candidate recommendation². While XForms is a relatively new specification, it is based entirely on established XML technologies (XML, XPath & XML Schema) and has been designed to work in conjunction with technologies such as AJAX. XForms can be embedded directly inside regular web pages (XForms+HTML) or separated out into different files.

With XForms, all data is represented as XML documents which can be stored in a file, a database or an external web service. Form designers create an XML document, the model, to encapsulate the data to be stored. On top of this is a set of XPath expressions that define the type (format) of data and any constraints placed on this data, such as requirements or dependences. The final layer, the interface, is a regular (X)HTML document with additional XForms elements. These elements reference parts of the XML data document - they are placeholders and are used to specify where in the web page parts of the XML model should be exposed. At run time an XForms engine converts the embedded placeholder elements into regular form input controls (<input>, <select>, etc.). This three-layer model isolates the data being stored from the end-user interface. Such a deliberate separation means that XForms can help improve authoring, reuse, internationalization, accessibility, usability, and device independence. For instance, the form view could be rendered as a Java or Flash application.

Although XForms has many attractive features, it is not without limitations. Firstly there is a steep learning curve in understanding the technology, which is far removed from writing traditional <FORM> elements in basic HTML pages. In addition to understanding the approach, there are very few editors or IDE supported tools for writing XForms (although it is possible to use OpenOffice to generate XForms). Secondly as a new and still evolving standard, support for XForms is limited. The current popular browsers do not natively support XForm-enabled applications; however browser add-ons can be installed to add this functionality. This arrangement was not suitable for our needs as it places additional onus on the user to install additional software components (which might discourage them from using our application). We also faced the possibility of users accessing the site through university machines, which may be operating under heavily restricted user accounts. An alternative approach to running XForms is to run the XForms engine on the server. In this configuration a translation process converts the XForms pages stored on the server into HTML+JavaScript pages, which are then sent to the browser. Along with the main benefit of working with all the latest JavaScript-enabled browsers, having a server-side XForms engine also allows XForms applications to request data from external servers thereby safely circumventing the security restrictions placed on scripts running solely in the browser.

There is a lot of commercial and government interest in XForms. The number of XForms implementations is steadily growing. On the client-side, there are add-on products available for all the popular browsers to render XForms pages. Native browser support for the full XForms specification is still missing and this is, in our view, a key limitation to the adoption of this technology. Mozilla Firefox does have an ongoing XForms project which, it is proposed, will eventually become part of the core browser engine. On the server-side, IBM has an XForms-enabled server, Yahoo! Mobile Widgets will be implementing XForms for writing their widgets and the Google mash-up editor, although it doesn't implement the specification, appears to be heavily based on this technology³. There is also a desktop application, XFormsPlayer, for running XForms pages outside of the browser directly on the desktop.

Our chosen solution uses Orbeon Forms⁴; a Java-based server-side Open Source XForms engine. Orbeon Forms is based around the XForms 1.1 specification and supports all the current popular JavaScript-enabled browsers, including Internet Explorer, Firefox and Safari. On top of Orbeon Forms we run a Tomcat JSP application to handle the user interface and an Open Source relational database at the back end to store all application data.

Using the XForms approach in our application, means we can store all case data as XML and use the XForms engine to handle constraint checking, enforce data types and implement complicated

² XForms 1.1 W3C Candidate Recommendation <http://www.w3.org/TR/xforms11/>

³ <http://www.orbeon.com/blog/2008/01/08/yahoo-goes-xforms/>

⁴ Orbeon Home Page <http://www.orbeon.com/>

interface mechanisms such as repeating elements. This is especially important as our target audience uses large complicated forms and it then allows us to separate the form design process from the need to write any validation scripts. Another benefit is that the Orbeon XForms Event model harnesses AJAX technology, which means we can easily implement Web 2.0-like features such as automatically saving the progress of the form as the student is entering data.

4.1 Form Designer

The application we have developed, as mentioned previously, is called the Clinical Experience Recorder (CER). The CER provides teachers with the ability to build template web forms, which students fill in, select a tutor and then submit the form to that tutor for assessment. Assessed forms are returned to the student and reports deemed of sufficiently high quality can be deposited into a library of exemplar cases.

Within the schools involved in the EELS project, each school has their own set of case forms, and while most have a similar overall structure, there are specialised elements to each form which target the specific type of case being documented (human, dental, canine, equine etc). In our new system we needed a form-creation process that could be used by practitioners or support staff to quickly and easily create forms that mimic existing document layouts. The form editor needed to be graphical in nature and as easy to use as possible to support users who might have little computer expertise. To achieve this we developed a web-based form editor, written in JavaScript, which allows form designers to build template forms in a WYSIWYG environment using tabs, grouped elements, sub groups, repeating items and various input types (Figure 2).

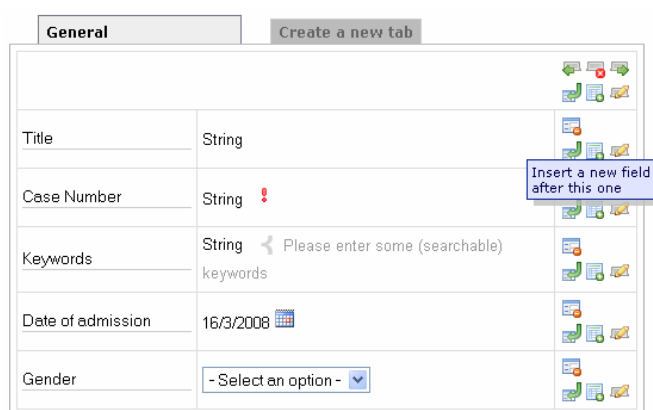


Figure 2 - Screenshot of the WYSIWYG Form Editor

The designer also has the ability to restrict access to field elements to specific user groups. The advantage of XForms means that the form editor is actually just generating a piece of XML, but behind the scenes at runtime our application will parse the XML and generate the relevant XForms presentation layer objects including a set of pre-defined XForms binding rules that apply all necessary validation. This XFORM+HTML page is then sent to the XForms editor to apply the final parsing into HTML+JavaScript.

4.2 Separation of Concerns

The form designer also doesn't need to worry about creating web pages, formatting the presentation or validating the user input as the CER application and XForms implementation work together to provide all this underlying support.

```
<xf:bind nodeset="instance(xmlDoc)/content//*[ @type = 'integer']" type="xforms:integer"/>
```

Figure 3 - Example of an XForms binding statement which applies integer validation rules to all elements of the XML document which have an attribute called 'type' set to the string 'integer'

Once a form template has been designed and built, it becomes available to students to use. Students instantiate the form by creating a copy - we call this instantiated copy a *report*. The report is a copy of the template's XML model which is then dynamically rendered into the HTML pages by the XForms engine and the CER. The student can fill in the report at their own pace and when completed, it can be sent to a tutor for assessment.

General	
Title	<input type="text"/>
Case Number	<input type="text"/>
Keywords	<input type="text"/> Please enter some (searchable) keywords
Date of admission	16/04/08 <input type="text"/>
Gender	- Select an option - <input type="button" value="v"/>

Figure 4 - Example of the instantiated form created from the Form Editor in Figure 2

4.3 New Interactive Form Elements

Included with the regular form element types (integer, rich text, dates, etc), we also provide a file input object which lets users attach files to their form. These can be text documents, pdf, images, sound or video segments. Another new data type we developed is a drawing object which can be used to allow users to annotate an image or create a new drawing from scratch. We provide this functionality through a relatively new HTML `<canvas>` element, which was introduced by Apple and can now be supported in all major browsers⁵.

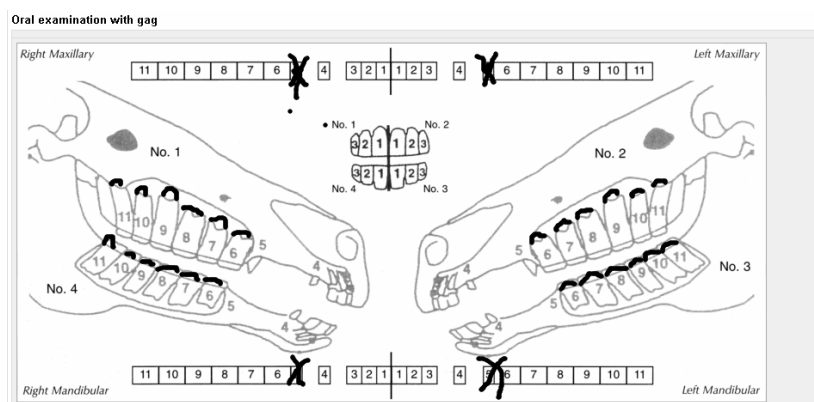


Figure 5 - Example of an annotated image using the Canvas object

4.4 Auto-saving, Auto-indexing and Auto-versioning

The XForms event model allows for AJAX-style interactions such as showing dynamic information based on changes to the XML data model (usually as a result of user actions such as filling in a field, or clicking a checkbox). We leverage this by executing a save action every time the user enters a new value into the report and changes the underlying data model (essentially mimicking Google Mail's auto-saving functionality). The XML model is then asynchronously sent back to the server and stored in the database. At the same time as the report is saved, we perform two additional actions. Firstly we extract the contents from every element in the XML model to store as an additional database table field, which we use when performing simple report searching. Secondly we check for the differences between the current report and last saved state in order to record the changes.

⁵ <http://developer.mozilla.org/en/docs/HTML:Canvas>

These can then be used to provide a complete history of the report and to visualise the changes over time.

4.5 Customisable Workflows

Within the EELS project brief was the requirement to develop an application with appeal to a wide sector of the university. We already had Veterinary, Dentistry and Medical schools involved in the project and could see wider uses for other faculties such as Engineering and Arts. The teaching methods used by different faculties are varied and adapted to meet discipline needs. Sometimes students work in groups on a report, and sometimes on their own. Sometimes the report is submitted to a tutor for initial review and then returned to the student for further editing, and at other times students complete the whole case before entering a single assessment phase. To cope with such different approaches, we needed to develop a flexible workflow engine that could be adapted to suite different requirements. The workflow engine then handles the progression of student reports through various stages. For example, the case library, where exemplar reports are archived, is represented in the workflow as a stage for non-editable reports which are visible and searchable by anyone.

The workflow is specified as an XML document which documents the various stages of an instantiated report and the actions and access rights of users to that report in each of the stages. Included in the rights management section are rules governing which user groups can see a report, edit a report, search for reports, review a report or move it through a transition to a new stage. On transitions between stages, actions can be defined to save a copy of the report, change the state of a report or send an email to a user. A workflow is assigned to a specific user and the CER can manage different workflows for different user groups. The flexibility provided by the workflow engine means we can implement any number of review stages (with different cohorts), submission-review and re-submission cycles, support collaborative student group work, and multiple assessment and archival phases.

4.6 Inline Feedback and Discussion

To support review and assessment, we wanted to mimic Microsoft Word's ability to add inline comments; the mechanism currently favoured by tutors. We enable this comment facility via a workflow permission. If the user has sufficient rights, then they can insert small comments next to any field on the report. This comment is actually being stored as an additional (`comment="xyz"`) attribute on the XML data structure. When the report is then viewed by the student these comments will appear as pop-up items next to the relevant field.

Alongside assessed feedback, the power of a student-teacher dialogue within an educational context has long been valued. To support this we provide a discussion-like mechanism which allows both students and tutors to discuss a particular report. Again permission to view or contribute to these discussions is restricted by the workflow engine.

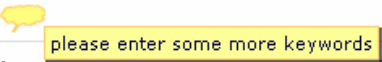
General	History	Physical Examination	Results and interpretations
Title	Nelson George		
Case Number	07.013418		
Keywords	FIP 		
Cat's name	Nelson		
Age of animal	4 years		
Breed	Maine Coon		
Sex	Male (neuter)		

Figure 6 - Example of a pop-up comment

4.7 Multi-facet Search and Browse

As the number of reports builds up in the system there becomes more need to search and retrieve reports. The CER provides a single search box interface to find cases based on the text entered; this is conducted on a separate database field which contains all the content of a report's XML elements (extracted when the report is saved). While useful for text strings, integer or date data is more difficult to locate using this method so we also provide a more advanced search mechanism whereby users can drill down to find specific reports. They select a report type and are presented with a blank template where they can enter values into the fields (for instance selecting a specific date or choosing an option from a select box). This process allows queries such as *'find all reports documenting male patients'* or *'find all reports documenting 5 year old cats'* to be conducted.

4.8 Management Interface

For tutors and other managerial users, we provide a management section of the application where reports can be viewed. These reports can be in any workflow stage (providing the manager in question has view rights to the report) and can be filtered by author, report type, date or report stage (e.g. assessed reports, library reports).

5 EXAMPLES OF USE

The initial CER was targeted towards the capture, review and assessment of clinical case reports for supporting veterinary, dentistry and medical staff and students at the University of Bristol. These stakeholders all have the same basic requirements although their individual case templates and workflows differ. To date we have dentistry and veterinary staff and students using the CER for clinical case reporting.

In all the conducted trials, no formal training was given to either staff or students, although members of staff were introduced informally to the software through various meetings relating to the project. Where possible, workflow stages and report templates use domain-specific terminology to aid users. The software also has several pages of documentation, an example 'dummy' form to play with, a screen-capture video introducing them to the software and a feedback section where they can leave messages to the development team (feature requests, support requests, bug reports etc).

5.1 Trial 1: Vets (CER context)

Our first trial of the CER was delivered to veterinary students and replaced a sub sample of final year clinical rotation reports traditionally submitted via Microsoft Word. The system went live in April 2007 and was used by tutors both as an alternative to, and in some cases alongside, the existing reporting method. Three different report templates were created; Canine, Feline and Equine, and these were then completed by students on repeating three-week rotations. To encourage students to engage with digital case reporting, USB memory sticks were provided to each student to make it easier for them to capture images and other multimedia (there is a dedicated AV unit inside the Veterinary School where students can be supplied with xrays and other imagery).

There have been around 200 students using this system over the last 12 months. Between them they have generated over 230 completed and assessed case reports.

5.2 Trial 2: Dentists (CER context)

In the autumn term of 2007 a second trial started with staff and students in the School of Dentistry. The dentists have similar reporting needs to that of the vets but use a slightly more complicated workflow incorporating submission, review & comment and resubmission loops. They also face an issue with documenting patient consent in a digital medium. Their chosen solution was to use the file upload facility to have students attach scanned copies of consent forms to their reports. Students within the final year of the Bachelor of Dental Surgery programme document their final assessment cases in Restorative Dentistry using the system. Cases are shared with designated

mentors for comment prior to being made available for assessment.

In this trial students have generated over 60 reports.

5.3 Trial 3: Dentists (ePortfolio context)

Our second use case is the capture of student competence. In this scenario, students fill out a form describing each type of skill they have observed or attempted. This report can then be viewed by an observing clinician and the unit organiser, and is stored in the system for later use as a basis for building up a portfolio of a student's clinical experience.

The Dentists have already run a trial of about 60 students in this mode. Each student uses the system to record their hours spent observing various clinical procedures and providing reflective feedback on their progress.

6 REACTIONS AND FEEDBACK

6.1 Staff and Student Feedback

Students have really been engaging with the CER. Most reports have attached pictures, videos, xrays and scans. Some have even used their mobile phones, unprompted by their tutors, to capture images of their cases, which they then have then uploaded. Veterinary students not engaged in the pilot, have enquired when it will be available for all to use. Feedback received from 7 of the veterinary tutors has been overwhelmingly positive. Tutors feel the software is reasonably intuitive, and all mentioned they had positive experiences. Tutors are aware that flexible online case reporting such as the CER provides is becoming increasingly essential. The general consensus among the tutors is that using the CER has led to a perceivable increase in quality of the reports submitted by students. Reports are more extensive and better documented than before however this quality comes at a cost which is largely carried by the tutors. With larger reports and instantaneous submissions, also brings expectations of higher quality feedback from tutors. We received comments such as:

'I think it is probably beneficial for the student, but it slows the assessment process. More detailed feedback is good'

'Slower than conventional marking, but a more complete report from the student'

In the dentistry trials, students were asked to comment on their experiences of using the Clinical Experience Recorder and subsequent implications for ePortfolios. The majority of students found the system relatively easy to use and 88% responded that it would be useful to have their own ePortfolio. When questioned on providing exemplar student reports in an online library for students to use as a learning resource, 100% of the respondents thought this is a good idea.

The highly positive empirical feedback above will be tested by analysing and comparing students' results, with the intention of providing more substantive evidence of the impact of the software on student learning and achievement.

6.2 Technical Feedback

Although the software was promoted as being part of a beta trial, and users were asked to report any problems encountered, the level of bug instances has been incredibly low. While the software has occasionally suffered from performance-related issues, partly due to the lack of a dedicated server and partly due to the heavy amount of AJAX traffic generated between client and server, there have only been a handful of bugs identified by users in the past year. This level of reliability, while mostly unnoticed by users, gives us strong confidence in the technical quality of the software.

7 CURRENT STATE OF DEVELOPMENT AND FUTURE WORK

The CER has now completed the first phase of development and deployment. Our trials have been

ongoing for over 12 months and a second version with an extended workflow model and minor bug fixes has just been released. This newer edition has a finer grained permission control allowing individual fields on a form to be editable to different users based on their role. This version is being used in an ePortfolio context where a student records a list of procedures and practices they have observed, then when they submit the report to their tutor, the tutor can see the student's data but along side have access to a different set of fields where they can record their opinion of the student's competency in each procedure.

In addition to incorporating feedback from staff and students in subsequent releases, two further phases are planned, the first of which is to provide more tools to aid in the analysis of reports. Currently reports can be searched using keyword terms or via an advanced search facility with fields containing specific values. Reports can also be viewed by tutors and filtered by author, report type, workflow stage, or assigned tutor. This provides a reasonable amount of flexibility and control to view a student's report however each report can only be viewed or printed out. We would like to extend this functionality to analyse the data contained in reports pull out specific (matching) information and generate meta-reports. This would be useful, for example, in situations where a patient has been seen by multiple students, leading to more than one report existing about that patient. It would be very useful to generate a consolidated report about the progress of that patient as seen in the different student reports. In an ePortfolio context, where a student is providing multiple competency assessment reports, it would be advantageous if we could provide a single overview of that student's competency from analysing all the different reports they have submitted.

As the trials continue, feedback has been very positive but we are nearing the end of the initial project funding. We are keen to expand our trial out to other departments within our university such as engineering or humanities. We feel the flexibility offered by our solution means there are many more potential uses of this software that we haven't even been able to envisage yet. As technology-supported learning becomes increasingly popular, universities (including our own) are keen to look for possible software solutions to offer ePortfolio solutions to students. The software developed for the EELS project has a lot of potential to offer in this role and we would be keen to see it used in the wider ePortfolio role.

7.1 Availability for others

The CER has been demonstrated to other academics and practicing veterinary clinicians in the UK, Europe and Australia and this has generated considerable interest. ILRT and the University of Bristol are currently exploring the best means to provide the software as hosted service.

There has been discussion within the University and with other Veterinary centres of learning of deploying the CER as a UK-wide national service available to all Veterinary Schools.

7.2 Access to the Software

The Clinical Experience Recorder is located at: <http://eels.ilrt.bris.ac.uk/cer>. You can log in as a student with the username/password: `student/student` and as an example tutor with `tutor/tutor`. There is a support section of the website which contains documentation and a video walkthrough of the system.